VINCA ProFlow：一种支持多租户的 SaaS 业务流程系统

李厚福 孙君意 马强 王蕾 赵卓峰

摘要 业务流程作为一种支持业务协同的有效方式，在企业内部精细化管理、上下游合作伙伴整合中起着重要的作用，使决策层能实时掌握经营数据。但建设业务流程系统往往需要在软硬件采购、人员配备方面投入不菲的成本。本文针对当前中小企业在建设业务流程管理系统中存在的“需求”与“投入能力”之间的矛盾，提出了一种支持多租户的 SaaS 业务流程系统——VINCA ProFlow，强调利用云基础设施的能力提供流程应用托管和运行环境，为中小企业提供一种可按需租用的流程服务（BPM-as-a-Service）。各个租户在共享同一流程服务实例的同时，又可专享独立的数据空间，保证了租户间数据的有效隔离和隐私保护。文中有针对性地讨论了 VINCA ProFlow 的研发动机、关键问题以及求解方法。

关键词 云计算 软件即服务 多租户 业务流程软件服务

1 引言

流程管理技术可以为集成型应用提供支撑环境，帮助企业、单位对业务流程获得可见性并实施控制，广泛应用于办公、物流、科学计算、应急联动等领域。但由于传统流程管理软件系统的实施成本（包括服务器硬件和流程管理软件的采购成本以及相应技术人员的人力成本）居高不下，使得众多正在创业期的中小企业对之望而却步。另一方面，有些企事业单位对流程管理的需求可能突然发生在某一个特定时期内，需求的临时性与高投入使其不愿意投入流程应用的建设，宁可靠人工执行流程，造成多个相关组织机构间数据延迟和不一致，降低了应对突发事件（如抗震救灾）的能力。

为了降低企业使用流程系统的成本，在行业云资源中心 VINCA TARC 中提供业务流程服务能力成为一个关键问题。为此，基于云计算技术 [0] 和 SaaS[1]模式，本文提出了一种支持多租户的 SaaS 流程系统——VINCA ProFlow，目标是提供云计算环境中可按需租用的流程服务（BPM[2] as a Service）。基于 SaaS 模式，运营商可以集中控制 IT 基础设施、软件系统、运营以及升级维护等成本，利用规模效应降低软件的运营成本，从而降低了每个客户的总体拥有成本。

2 相关工作

随着云计算的兴起，基于云计算环境的 SaaS 流程系统也开始出现。本节将首先对 SaaS 流程系统的现状进行简要说明，然后介绍相关关键技术的研究进展。

2.1 SaaS 流程系统

云计算是一种建立在抽象的、可共享和具有弹性的资源之上的 IT 服务消费和交付方式，而 SaaS 交付模型各种云计算使用模式中的一种。SaaS 应用部署并运营在供应商的数据中心，并允许多个终端用户远程访问同一个应用实例。SaaS 交付模式的一个特点是其租赁模型，

[1] Software as a Service，软件即服务（或软件作为服务）
[2] Business process management，业务流程管理

目前 SaaS 供应商采用的租赁模型大致可归纳为如下三种。

表 1. 三种不同的租赁模型

租赁模型	描　　述	优缺点
单独租赁	不同租户拥有不同的应用实例，每个实例对应特定的数据源；	数据独立性好，但需要供应商提供更多服务器空间，成本较高
全共享租赁	所有租户共享同一应用实例，所有租户的数据保存在相同的数据库中，通过逻辑方式进行数据隔离；	对运营商和租户而言成本都较低，但牺牲了数据的隔离性
数据独享租赁	所有租户共享同一应用实例，每个租户独享数据空间，实现了租户间数据的物理隔离。	成本介于单独租赁和全共享租赁之间，但保证了数据隔离性

目前 VINCA ProFlow 的 SaaS 租赁模型采取的是数据独享、应用实例共享的模式。除非 SaaS 供应商公开说明，不然从外部很难区分全共享租赁和数据独享租赁。

伴随着云计算的兴起，基于云计算环境的工作流产品也开始出现。例如，IBM 的 BPM BlueWorks[3]，富士通的 Cloud BPM[4], TIBICO 的 Spotfire 等。IBM BPM BlueWorks 提供了一系列基于云计算的业务流程管理服务，借助 BPM BlueWorks，企业领导者、企业分析师和企业专业人员将可以通过浏览器进行内容创建、共享和协作，充分利用预置的流程模板和来自流程管理专家以及全球用户的贡献，快速完成从制定战略到执行流程的全部工作。富士通的 Cloud BPM 提供了多租户支持，并且可以根据租户的需求对工作流的功能进行裁剪。TIBICO 的 Spotfire 除了提供基本的流程管理服务外，还具备业务活动监控和关键数据分析的功能。

除了工业界推出的这些云工作流产品，学术界也开始越来越多地关注基于云计算环境的工作流中存在的研究问题。这其中既有利用云计算的基础设施来改善原有工作流引擎性能的研究，也有探索在云计算环境中如何设计工作流架构方面的研究。美国印第安纳大学霍法（C. Hoffa）等人探索了科学工作流 Pegasus-WMS 在本地集群和云计算虚拟集群中的性能对比[5]。德国马尔堡大学的德纳曼（T Dörnemann ）等人开发了一款 BPEL[3]引擎插件[6][7]，可以将 BPEL 流程中的某些计算密集型活动动态调度到 Amazon EC2 的云节点中。加拿大多伦多大学的穆都萨米（V Muthusamy）等人探讨了云工作流的架构挑战，并提出了一种基于事件的、服务等级协议（SLA，Service Level Agreement）驱动的架构[8]，将流程动态分解成有发布/订阅(Pub/Sub)关系的多个活动，以适应云计算的分布环境。可见，基于云计算环境的工作流正逐渐成为工作流领域人们关注的焦点。

尽管工业界和学术界已经开展了诸多实践和研究工作，但对于如何定义 SaaS 流程系统还未形成定论。这是因为流程系统本身包含了流程发现、建模、仿真测试、规则、监控以及工作流引擎等多个部分，尽管目前业务流程管理开发商都已经认识到以 SaaS 模式交付流程系统带来的好处，但对于流程系统中哪些部分应该以 SaaS 交付并没形成一致。对不同厂商而言，SaaS 业务流程管理的内涵并不一样。因此，本文也不打算给出一个明确的定义。但从总体上看，目前国外厂商的 SaaS 服务以提供流程和业务规则建模为主，如 IBM（Lombardi Blueprint 和 Blueworks），Software AG Alignspace；有少数厂商具有以 SaaS 提供流程执行环

[3] Business Process Execution Language，业务流程执行语言。一种基于 XML 的，用来描写业务过程的编程语言

境的能力，如 Appian[4], Cordys(Cordys Business Operations Platform)和 Vitria；只有极少数厂商能够同时以 SaaS 提供流程建模和运行环境，如 PegaSystems[5]和 Adeptia BPM[6]。这些厂商采取的租赁模型也不尽相同，如 Adeptia BPM 采取的是单独租赁模式，即为每个租户提供一个或若干虚拟机来部署其业务流程管理实例，而 IBM 则采用共享应用实例的模式。

2.2 用户端与云端的业务数据联动

正如在第一节中所论述的，实际应用中，脱离用户端的“纯”云工作流是不现实的。因为用户端往往具有满足特定需求的业务系统，即使只是一般的办公软件，其中也会维护着重要的业务数据。为了实现用户端与云端业务数据的实时联动，我们调研了数据同步的相关技术，尤其是云计算环境下的数据同步技术。

数据同步技术是一个经典的问题，早在 1996 年特里戈尔（A. Tridgell）和马克拉斯（P. Mackerras）就提出了 rsync 算法[9]。该算法采用了增量同步的技术，已经被广泛用于各 Unix 发行版。随着分布式文件系统的成熟和云存储服务的出现，有一些利用云计算环境来实现数据同步的研究开始出现。澳大利亚墨尔本大学的布罗伯格（J. Broberg, ）等人开发了 Cloud Storage Mashup[10]，可以集成 Amazon S3、Nirvanix 等云存储服务，实现高性能、低花费的数据分发和同步。国防科大茅华建（音译， Huajian Mao）、美国韦恩州立大学施巍松（Weisong Shi）等人开发一款名为 WuKong[11]的面向云的数据服务，可以借助云实现 PC 以及其他手持设备的实时同步，其特点在于提供了一个抽象层，透明地集成各种云存储服务。

数据同步技术的研究由来已久，在云存储逐渐发展起来之后为这一技术的进化提供了新的可能。当前已经出现了一些利用云存储实现数据同步的研究工作，但是主要还是通过接口集成，缺少在新的背景下的算法创新。

2.3 工作流数据管理的可扩展性

基于云的工作流面向互联网范围提供服务，因此其处理的流程请求的密集程度、数据的规模都远远高于传统的工作流系统。在请求密集、数据量大的场景中，流程数据的读写往往成为系统的瓶颈。随着类 Google BigTable[12]的一些分布式数据存储系统的出现，一些研究者开始考虑使用开源分布式数据存储系统如 HBase[13]等来增强流程管理系统的可扩展性。

美国加州大学的王建武（Jianwu Wang）等人研究了将科学工作流软件 Kepler 与 Hadoop 集成，减少了有关流程数据处理的时间且改善了 Map-Reduce 的易用性[14]。加拿大滑铁卢大学的章陈（音译，Chen Zhang）等人在 Hadoop 的基础之上开发了一款轻量级流程管理软件 CloudWF[15]，使用 HBase 来存储流程变量，使用 HDFS 存储业务文档，使得系统具有高可扩展性。美国德州大学的阿布拉罕（J. Abraham）等人采用了 HBase 来存储服务输出的 RDF[7] 数据，并且优化了 HBase 的存储结构以提升 SPARQL 语言的查询效率[16]。

然而，目前的工作流数据的可扩展管理机制研究主要集中在科学工作流领域，还缺少通用工作流系统的数据扩展方案，而通用工作流系统对于数据一致性、可定制的查询要求更高。

3 关键问题

[4]Appian:http://www.appian.com/registerUserWebinarBpmit.do
[5]PegaSystems: http://www.pega.com
[6]Adeptia BPM:http://www.adeptiabpm.com/?gclid=CIewp5ep0aQCFc8EHAodz2P_LA
[7] Resource Description Framework，资源描述框架

基于云计算环境的、可租用的 SaaS 流程系统使用模式与以往相比，发生了很大变化：由以往的一家独享变为多家共享；流程系统不再部署在企业内部，而是部署在第三方运营中心；系统的用户数和数据量由可预测变成难以预测；服务质量保障不再是自己说了算，而是受到服务等级协议的约束。这一系列的变化，对于如何在行业云中心提供可用的流程服务带来了很多技术上的挑战。这包括：如何为用户提供基于浏览器的流程建模工具，以便支持多用户协同建模；如何保证用户数据不被其他租户偷窥和窃取；如何在某些服务器出故障的情况下仍能保障租户业务的可持续性；如何提高系统的可伸缩性，以便满足持续用户和数据的持续增长；如何支持用户本地业务系统与云端流程系统的集成；如何支持云环境下跨组织的业务协同。这些问题有些在相关工作中得到解决，但有些仍然没有找到好的解决方案。限于篇幅，这里只选择了与本文主题密切相关的两个问题进行讨论，以部分说明 VINCA ProFlow 的研发动机。

3.1 流程系统中的数据管理

SaaS 流程系统中的数据管理与传统流程系统中数据管理的区别主要体现为三个方面：

1. **数据隔离存储** 传统流程系统中的数据存放在企业自身服务器，不暴露在公共的互联网上，不必过多担心其安全性。但在 SaaS 模式的流程系统中，为了避免业务数据外泄，用户希望其流程数据与其他用户数据的隔离有可靠保证。

2. **用户端与云端数据联动：**由于跨管理域的特点，基于云计算环境的工作流需要处理的业务数据可能分布在不同的自治域中。流程引擎要访问用户端的业务数据有两种方案：一种是用户端将业务数据封装为服务供云端调用，但这对于用户端的 IT 建设水平要求较高，且容易形成单点依赖，甚至造成 Slashdot 效应[8][2]；另一种方案是通过数据同步技术实现用户端和云端业务数据的联动，将用户要共享的业务数据实时汇聚到云端，供流程引擎及流程参与者访问。其中的技术难点就是如何保持云中的数据是相对“新鲜”的。

3. **数据管理的动态扩展：**工作流系统提供的是一种有状态的服务，对业务流程的状态的管理依赖于工作流数据的管理，频繁的状态变迁将引起工作流数据的密集读写。传统工作流多采用关系数据库管理工作流数据，这种集中式的数据管理方式能够满足单个企业或组织内部流程管理的需要。然而，基于云计算环境的工作流系统要面向整个互联网提供服务，传统的流程数据管理方式已经难以适应数据管理的高可扩展、高并发读写等需求，虽然目前已经有了诸如 BigTable[12], Amazon S3, HBase[13]等基于云计算环境的数据管理软件，但是这些软件提供的数据查询服务还较为单一，以键值查询为主，能否满足流程系统中数据管理需求还有待进一步的研究。

3.2 可伸缩、高可靠的流程运营支撑

SaaS 系统具有用户量大、用户数量持续增长、用户访问时段具有周期性和“扎堆”的特点，这对流程应用的运行支撑提出了很高的要求，系统必须能够在用户请求集中时提供更多处理能力以避免响应时间的明显增加，并在用户请求降低时去掉一些处理能力以节省运营成本。此外，在关键性的业务应用中，在流程执行过程中业务流程引擎出现异常或宕机时，应该提供相应的异常处理机制，比如将异常引擎中的流程实例转移到健康的流程引擎上继续执行，使用户的业务能够顺畅地进行。因此，作为一种可持续运营的流程 SaaS 系统，必须考

[8] Slashdot 是一个网站，每天都会更新在主页的新闻数次。较受欢迎的新闻可能会有上千的浏览者发表意见。Slashdot 效应是指当成千上万的读者连结到某一在 Slashdot 公布的网址时，导致该网址因访问人数激增而瘫痪的效应。

虑到运营期间系统用户和数据规模的持续增长，并且必须为用户业务的可持续性提供有力保障。为此，需要提供一个可伸缩、高可靠的流程运行环境。

4 VINCA ProFlow

4.1 目标

基于以上分析，VINCA ProFlow 的设计宗旨是针对中小组织机构自身 IT 设施落后的现状，为中小型组织机构提供一种可按需租用的流程服务（BPM-as-a- Service）。具体目标为：

1. 支持数据独享租赁模型：各个租户在共享同一流程服务实例的同时，又可专享独立的数据空间，保证租户间数据的有效隔离，保护租户隐私。

2. 可扩展的数据管理：当 VINCA ProFlow 现有的硬件资源不足以存储流程数据或支撑高并发的流程数据访问时，可以通过增加新的服务器来分担存储和数据访问的压力。这一过程对流程引擎应该是透明的、平滑的，不需要修改代码或停止正在运行的流程实例。

3. 支持用户端业务系统与云端应用的数据集成：为使中小组织机构托管在云端的业务流程能够使用到组织机构在本地的"最鲜活"的业务数据，需要将本地业务数据的任何微小变动实时同步到云端业务数据空间，供业务流程执行时使用。

4. 提供可靠、可伸缩的流程运行环境：可靠是指在引擎服务器出现故障时系统能够将故障服务器上的流程实例在较短的时间内迁移到正常服务器上继续执行，从而保证用户业务的可持续性；可伸缩是指当用户并发访问密集时，可以动态增加引擎服务器，从而水平扩展系统处理能力。而当用户访问量下降时，又可以逐步减少引擎服务器数量，以降低成本。

4.2 关键技术

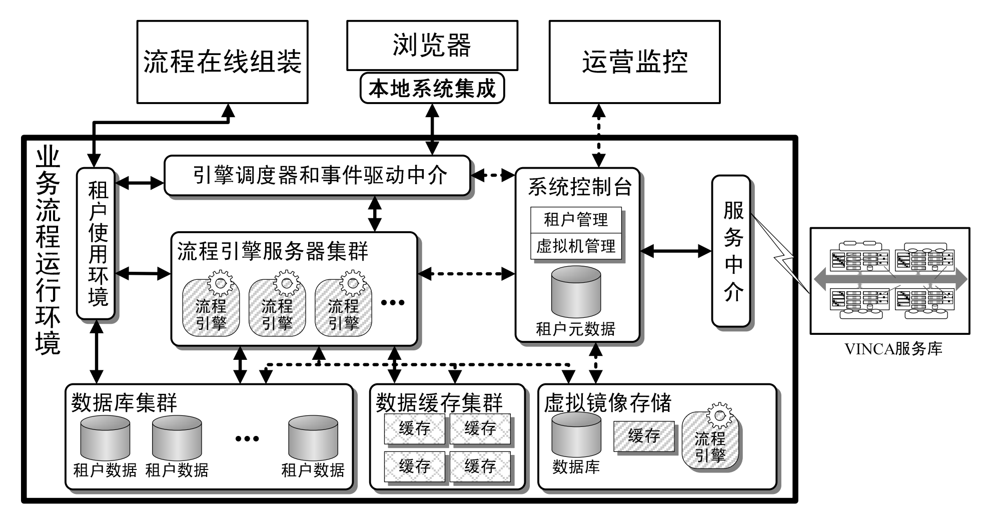


图 1. VINCA ProFlow 的组成

为了实现上述目标，本文在前期工作的基础上做了两方面工作：首先，在 VINCA ProFlow 的体系结构中引入了多租户共享实例、独享数据空间的 SaaS 租赁模型，实现了租户间组织机构数据、流程模型数据、流程运行时数据以及个性化配置数据的隔离存储，能够避免租户数据的被偷窥和窃取；其次，通过引入虚拟机管理、引擎集群管理和调度、数据管理等机制，提供了一个可动态伸缩、可靠的业务流程执行环境，能够应对用户请求和数据的不断增长和

急剧变化，如图 1 所示。限于篇幅，本文仅选其中最相关的几项关键技术进行汇报。

1. 支持多租户的、可扩展的数据管理技术

为了达到§4.1 中提到的前三个目标，在 VINCA ProFlow 研发过程中，集中开展了面向多租户的数据管理动态扩展技术。该技术从如下三个方面取得了一定进展。

– 基于虚拟数据服务的多租户数据隔离

在多租户的架构下，如何保障租户的数据隔离是一个具有两方面困难的难题。一方面是数据的访问权限的隔离，即租户在未经授权的情况下不能访问彼此的数据，以免泄漏数据隐私；另一方面是数据访问的服务质量的隔离，即当一个租户提交一个慢查询后，其他租户的数据访问请求不会因为数据库服务器太忙而被延时或拒绝。虚拟机技术是实现资源隔离的有效方法，VINCA ProFlow 通过 Xen[9]把安装有关系数据库、键值数据存储系统的操作系统封装成虚拟机镜像，在每个虚拟机实例上，在不同的端口运行着多个数据服务进程，每个租户独占自己的数据服务进程。VINCA ProFlow 在创建租户时能够为其自动创建其应用需要的虚拟数据服务，用来保存其流程数据、组织机构以及个性化配置数据，实现了租户间的数据隔离，而且可以通过图形化的监控仪表面板查看系统中各租户对配额的使用情况。

– 基于数据分片的用户端-云端业务数据联动

在实现用户端业务系统与云端系统集成时，最关键的是实现本地数据与云端数据的联动。在实际应用中，业务数据的变更往往体现出局部性特征，用户端更新的数据与云端的数据副本只存在部分差异。为此，本文提出了一种基于数据分片的用户端-云端业务数据联动机制，旨在快速识别出数据差异，减少数据流量，从而提高业务数据联动的实时性。其工作原理是：

在用户端部署代理程序，通过注册文件系统钩子（hook）以及监视数据库日志等手段实现事件驱动的数据变化监测机制。代理程序负责根据业务数据类型将其划分为一定粒度的片段，并计算每个片段的特征值。对于纯文本的文档数据，以一定的行数为分片的单位；对于二进制格式的文档数据，以一定长度的字节窗口为分片的单位；对于关系型数据库管理系统中的数据，以一定行数的元组窗口为分片的单位。数据特征值的计算可采用弱校验和强校验结合的算法。当用户端数据被修改后，用户端代理将业务数据的特征值传输到云端，云端通过比对找出缺少的数据片段的特征值，并将这些特征值返回给用户端的代理。然后，用户端代理将云端缺少的数据片段压缩后发送到云端，云端将这些数据解压后归并到相应的位置。

– 基于特征的流程数据的可扩展管理

传统业务流程系统一般采用关系型数据库管理流程数据，但是采用关系型数据库存储流程数据在可扩展性上存在局限。关系型数据库的经典可扩展方案主要有两种：读写分离[17]和数据分片[18]。读写分离适合读多写少的应用，在数据更新频繁的环境下数据延迟的情况会很严重；数据分片技术虽然提高了可扩展性，但是牺牲了关系数据库的事务保障、复杂查询等功能。近年来在互联网计算领域出现的分布式键值存储系统（如 BigTable[12]、Dynamo、Cassandra[19]等）在基于键的读写性能上做了特殊优化，且能够通过对等网络（P2P）协议将数据在多台服务器上进行划分和复制，使得系统的水平扩展能力增强。我们考虑结合各种不同的数据管理工具的优势来进行流程数据的管理。通过初步的实验，我们观察到流程数据具有不同的访问特征，如表 2 所示。元数据的读取操作多于写入操作，查询语句复杂度不高。

[9] 一个由剑桥大学开发的开放源代码虚拟机监视器。

例如：获取流程的定义，查询某个机构下的用户；实例数据的读写操作比例相当，读写都很频繁，查询语句较简单，主要以单键查询为主，例如：根据流程实例的 ID 更改它的状态、根据流程变量的 ID 更新其值等；日志数据的写入操作多于读取操作，但查询语句较复杂，且多为统计类查询语句，例如：查询一段时间内执行耗时最多的 10 个活动。

表 2. 流程数据的访问特征

数据类型	读写特征	查询复杂度
元数据	读多写少	复杂查询较多
实例数据	读写频繁	单键查询为主
日志数据	读写可异步	复杂查询较多

经过对这些特征的分析和总结，VINCA ProFlow 提出一种基于特征的流程数据可扩展管理机制，其思想是针对各类流程数据的特征采用不同的数据存储方式，综合集成关系型数据库和分布式键值存储系统来解决工作流内部数据的可扩展性方面的问题。将关系型数据库用于元数据的存储，分布式键值存储系统 Cassandra 用于存储运行时的流程实例数据，利用分布式键值存储系统的可扩展性来增强工作流运行时的可扩展性。此外，通过抽象的流程数据访问层 (Process Data Access Layer, PDAL)为工作流引擎屏蔽了底层数据模型的差异，并在内部完成了对流程数据的适配和路由，也使得引擎的数据访问代码不需要随着节点个数的伸缩而修改。

2. 引擎集群管理和调度技术

为了实现§4.1 中目标 4 提出的可伸缩性，VINCA ProFlow 采用了分布式多引擎的架构。底层数据一致性由流程数据访问层提供。元引擎负载对多个流程执行引擎的调度和管理。流程执行引擎的数量初始为 1-2 个，当系统负载较大时，管理员可以通过系统管理控制台将相对空闲的流程执行引擎加入集群，分担系统的负载。当没有足够可用的流程执行引擎时，管理员可以激活安装有流程执行引擎的虚拟机镜像的实例，实现快速扩容。引擎集群的调度原理为：当元引擎接受到前端的工作流门户发出的请求后，根据其请求的应用，通过一级调度器的决策，选择一个流程执行引擎来执行其请求的流程。元引擎会记录每次流程请求的响应时间，然后周期性地调整各个流程的权重策略。此外，由于流程中不仅包含本地服务调用，还有远程服务调用，有时一个远程服务调用活动可以由多个节点提供，故每一个流程执行引擎有独立的二级调度器，用于调度应用执行过程中的服务调用活动。

– 基于流程相似度的权重预设

为了减少与新部署的流程对应的权重策略的权值收敛过程所需要的时间，我们提出一种基于流程相似度的权重预设算法。如果现有的系统中没有部署任何流程，那么新部署的流程对应的各引擎分配权重设置为均等；如果现有的系统已部署了一定量的流程，且经过一段时间的运行它们的权重策略已经收敛到一个稳定的状态，那么新部署的流程的权重策略的初始值就可以根据和它相似的其他流程的权重值计算出来。

– 基于流程特征的动态加权轮转

一般常用的调度算法有：简单轮转法、静态加权轮转法、基于请求数量的加权最小连接数优先法、以及动态加权轮转法等。结合 VINCA ProFlow 自身的业务特点，我们提出一种基于流程特征的动态加权轮转法。传统的动态加权轮转法对所有的请求采取统一的权重策略，而 VINCA ProFlow 采用的算法则是针对不同的业务流程使用响应采用不同的权重策略，不同的流程执行中对资源的消耗情况是不一样的，有的流程是计算密集型，有的是 I/O 密集型，各流程执行引擎所在节点的计算能力、I/O 吞吐能力和操作系统也不尽相同,因此需要为每种类型的流程维护合适的权重策略。

– 两级调度机制

由于流程调用的服务中有一些是本地服务，有一些则是其他节点上的服务，且同一功能的服务可能在不同的节点上有副本，当流程运行过程遇到这种情况时，就需要选择由哪一个节点来执行该服务。由于各个节点的软硬件资源的差异以及通信速度的差异，服务调用的响应时间也会有差异。因此，为了优化服务调用的响应时间，VINCA ProFlow 设计了服务二级调度机制，即除了元引擎有针对应用的一级调度器，每个引擎还维护一个针对服务调用的二级调度器。该调度器将远程服务调用活动的请求按一定的权重分配给部署了该服务副本的节点。

3. 引擎失效检测及流程实例迁移技术

要实现可靠的流程执行环境需要解决如下几个重要问题：

– 准确及时地发现失效引擎，并迅速迁移和恢复失效引擎中未完成的流程；
– 引擎间共享流程执行信息，以便在某一引擎失效时，其它引擎能够获取失效引擎中未执行完成流程的信息；
– 选择合适的流程恢复起点，避免重复执行已完成的部分。

针对上述问题，我们对流程引擎失效检测及实例迁移技术进行了研究，在引擎集群和元引擎管理的基础上做了三方面工作：

– 扩充了 WMFC（the Workflow Management Coalition，工作流管理联盟）规范中的流程活动运行时模型，以更细粒度描述了流程活动的执行过程；
– 采用分布式缓存的方法实现流程信息在引擎集群中的共享。当出现引擎失效时，其它引擎可以从缓存服务器中读取失效引擎中未完成流程的信息，并进行恢复；
– 采用组播技术检测引擎是否失效。当元引擎检测到失效引擎时，及时将失效引擎中的未完成流程实例迁移到健康引擎中并回复执行。

5 评价和验证

5.1 扩展性评价

为了验证 VINCA ProFlow 的扩展性，本文使用 JMeter、Apache Benchmark 等工具模拟高并发的流程请求场景，对该系统在不同部署模式下的吞吐率（单位时间内处理流程请求的个数）变化情况进行评价。

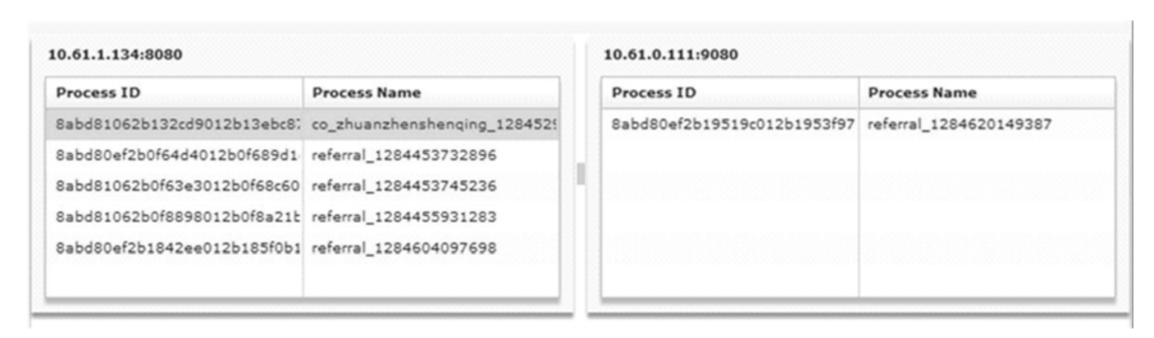
本文采用的测试案例流程请求总计为 100000 个，并发度为 1000。测试使用的流程是一个无人工参与的流程，包括了顺序、分支、循环等基本逻辑结构，流程中的活动为脚本活动，以避免调用 Web Service 引入外部系统的性能干扰。系统部署模式分别为：（1）一个引擎配一个数据节点；（2）保持一个数据节点不变，动态增加流程引擎的数量；（3）同时动态增加流程引擎和数据节点的数量。

测试结果表明：在部署模式（1）中，由于单流程引擎和单数据节点受到单机物理性能的局限，其吞吐率达到一定程度后很难再进一步提升；在部署模式（2）中，通过增加流程引擎的数量，可以进一步提升系统吞吐率，但最终仍会受到数据节点处理能力的限制；在部署模式（3），可以通过同时增加引擎节点和数据节点实现吞吐能力的不断扩展。

5.2 可靠性评价

为了测试 VINCA ProFlow 的可靠性，我们首先启动两个引擎，然后关闭其中一个引擎，观察运行在被关闭引擎中的流程实例是否被成功迁移到另一个引擎中并恢复执行。如图 2 所示，测试前系统中有两个健康引擎，IP 为 10.61.1.134 的引擎中有 5 个正在执行的流程，IP 为 10.61.0.111 的引擎中有 1 个正在执行的流程。

测试时，强制关闭 IP 为 10.61.0.111 的引擎。如图 3 所示，被关闭引擎中的 1 个流程被迁移到 IP 为 10.61.1.134 的引擎中继续执行。



10.61.1.134:8080

Process ID	Process Name
8abd81062b132cd9012b13ebc8	co_zhuanzhenshenqing_128452
8abd80ef2b0f64d4012b0f689d1	referral_1284453732896
8abd81062b0f63e3012b0f68c60	referral_1284453745236
8abd81062b0f8898012b0f8a21b	referral_1284455931283
8abd80ef2b1842ee012b185f0b1	referral_1284604097698

10.61.0.111:9080

Process ID	Process Name
8abd80ef2b19519c012b1953f97	referral_1284620149387

图 2. 集群中所有健康引擎的流程信息列表

10.61.1.134:8080

Process ID	Process Name
8abd81062b132cd9012b13ebc8760ce6	co_zhuanzhenshenqing_1284529439270
8abd80ef2b0f64d4012b0f689d1e026f	referral_1284453732896
8abd81062b0f63e3012b0f68c60301c2	referral_1284453745236
8abd81062b0f8898012b0f8a21bb0048	referral_1284455931283
8abd80ef2b1842ee012b185f0b140021	referral_1284604097698
8abd80ef2b19519c012b1953f9780008	referral_1284620149387

图 3. 关掉一个引擎后，健康引擎的流程信息列表

6 结束语

VINCA ProFlow 作为 VINCA TARC 的组成部分之一，目标是提供一种可租用的 SaaS 业务流程系统。本文首先基于中小企业对业务流程管理的需求与其投入能力之间的矛盾，分析了在 VINCA TARC 中提供流程服务能力的必要性；然后讨论了国内外相关工作进展，并分析了以 SaaS 模式交付业务流程系统时亟待解决的关键问题；最后讨论了中科院计算所自主开发的 SaaS 流程系统 VINCA ProFlow 的目标、关键技术以及试验评价。

参考文献：

[1] Brian Hayes. Cloud Computing. *Communications of the ACM*, 2008, 51(7):9-11.

[2] Slashdot Effect. http://en.wikipedia.org/wiki/Slashdot_effect.

[3] IBM BPM BlueWorks. https://apps.lotuslive.com/bpmblueworks/.

[4] Fujitsu Cloud BPM: Fujitsu Global. http://www.fujitsu.com/global/services/software/ interstage /solutions/bpm/cloud-bpm.html

[5] Christina Hoffa, Gaurang Mehta, Timothy Freeman. On the Use of Cloud Computing for Scientific Workflows. *In IEEE Fourth International Conference on eScience (eScience '08)*, Indiana, USA,Dec.

7-12, 2008, pp.640-645.

[6] Tim Dörnemann, Ernst Juhnke, Bernd Freisleben. On-Demand Resource Provisioning for BPEL Workflows Using Amazon's Elastic Compute Cloud. *In Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid* (*CCGrid '09*), Beijing, China,May 18-21, 2009, pp.140-147.

[7] Tim Dörnemann, Ernst Juhnke, Thomas Noll. Data Flow Driven Scheduling of BPEL Workflows Using Cloud Resources. *In IEEE 3rd International Conference on Cloud Computing (CloudCom '10)*, Miami, USA,July 5-10, 2010, pp.196-203.

[8] Vinod Muthusamy , Hans-Arno Jacobsen. BPM in cloud architectures: Business process management with SLAs and events. *In Business Process Management 2010 (BPM '10)*, NJ, USA.

[9] Tridgell, A. and Mackerras, P. The rsync algorithm. Technical Report TR-CS-96-05, Department of Computer Science, The Australian National University, Canberra, Australia, 1996.

[10] James Broberg,, Rajkumar Buyya, Zahir Tari. Creating a 'Cloud Storage' Mashup for High Performance, Low Cost Content Delivery. *In 6th International Conference on Service Oriented Computing (ICSOC '08)*, Sydney, Australia,Dec. 1-5, 2008, pp.178-183.

[11] Huajian Mao, Nong Xiao, Weisong Shi, Yutong Lu. Wukong: A Cloud-Oriented File Service for Mobile Internet Devices. *In Proceedings of the 7th International Conference on Services Computing (SCC '10)*, Miami, USA,July 5-10, 2010.

[12] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh. Bigtable: A Distributed Storage System for Structured Data. *ACM Transactions on Computer Systems (TOCS)*, 2008, 26(2):1-26.

[13] HBase. http://hbase.apache.org/.

[14] Jianwu Wang, Daniel Crawl, Ilkay Altintas. Kepler +Hadoop: A General Architecture Facilitating Data-Intensive Applications in Scientific Workflow Systems. *In Proceedings of the 4th Workshop on Workflows in Support of Large-Scale Science (WORKS '09)*, Portland, Oregon, USA, November 15, 2009, pp.1-7.

[15] Chen Zhang, Hans De Sterck. CloudWF: A Computational Workflow System for Clouds Based on Hadoop. In *CloudCom 2009*, LNCS 5931, M.G. Jaatun, G. Zhao, and C. Rong (eds.), Springer-Verlag, 2009, pp.393-404.

[16] John Abraham, Pearl Brazier, Artem Chebotko, etc. Distributed Storage and Querying Techniques for a Semantic Web of Scientific Workflow Provenance. In P*roceedings of the 7th International Conference on Services Computing (SCC '10)*, Miami, USA, July 5-10, 2010.

[17] Emmanuel Cecchet, George Candea, Anastasia Ailamaki. Middleware-based Database Replication: The Gaps Between Theory and Practice. In Proceedings of the 2008 *ACM SIGMOD International Conference on Management of Data (SIGMOD'08)*, Vancouver, Canada, June 9-12, 2008, pp.739-750.

[18] Pritchett D. BASE: An Acid Alternative [J]. ACM Queue. 2008, 6(3): 48-55.

[19] Avinash Lakshman, Prashant Malik. Cassandra: a decentralized structured storage system [J]. *ACM SIGOPS Operating Systems Review (SIGOPS Oper. Syst. Rev.)*, 2010, 44(2):35-4.

作者简介：

李厚福： 中国科学院计算技术研究所软件集成与服务计算研究分中心、助理研究员

孙君意： 中国科学院计算技术研究所软件集成与服务计算研究分中心、博士研究生

马　强： 中国科学院计算技术研究所软件集成与服务计算研究分中心、硕士研究生

王　蕾： 中国科学院计算技术研究所软件集成与服务计算研究分中心、硕士研究生

赵卓峰： 中国科学院计算技术研究所软件集成与服务计算研究分中心、高级工程师
zhaozf@software.ict.ac.cn